C4 Project

Cahier des charges

Aldéric Coroir (*coroir_a*)
Julien Freche (*freche_j*)
Romain Riedinger (*riding_r*)
Benoit Zanotti (*zanott_b*)

1. Introduction

Ce document a pour vocation de poser les bases et d'établir les besoins, objectifs et contraintes de notre projet de deuxième semestre. Après avoir introduit le sujet, nous présenterons notre projet, puis nous détaillerons chaque partie et nous terminerons par la répartition des tâches et le planning de réalisation.

Cadre

Le projet sera réalisé par un groupe de quatre personnes, il s'étendra sur environ cinq mois. Le sujet est libre, mais il doit cependant être validé par nos professeurs. Trois soutenances jalonneront l'avancée de celui-ci.

Le sujet

Le sujet étant laissé libre, nous avons choisis de réaliser un logiciel de reconnaissance optique de caractère, souvent appelés OCR pour Optical Caracter Recognition, ces logiciels sont capables à partir d'une image (typiquement un document scanné) d'en extraire le texte et de le convertir dans un format de fichier éditable.

Restrictions

Ce projet sera développé sous Linux (principalement sur Ubuntu 10.10 et Fedora 13), il sera codé en Objective Caml et en C. Il fonctionnera donc sur les deux systèmes précédents (et donc sur les machines du PIE), celui-ci devrait cependant marcher correctement sur la plupart des distributions Linux intégrant les paquets Ocaml et C nécessaires.

Protocole

Ce projet doit être réalisé par groupe de quatre personnes, dont un chef de projet qui servira entre autre d'interlocuteur avec le jury pendant les soutenances. Les groupes sont formés librement. Il est tout d'abord demandé de fournir un cahier des charges, afin qu'il soit validé.

Soutenances

Notre projet sera évalué lors de trois soutenances par un jury composé de deux de nos professeurs. Les deux premières soutenances auront une durée de 15 minutes, la dernière aura une durée plus longue. Lors de ces soutenances, nous présenterons clairement les avancées de notre projet par rapport au cahier des charges.

2. Nature et but du projet

Introduction

Notre projet sera un OCR (de l'anglais Optical Character Recognition), si vous possédez un scanner, vous pouvez numériser un document dactylographié et le logiciel se chargera de vous fournir un document texte éditable. Ainsi l'utilisateur a la possibilité de convertir ses documents papiers au format numérique afin de les archiver ou de les éditer et de pouvoir les (ré)-imprimer.

Principe

Le principe est simple : l'utilisateur fournit un document numérisé au format A4 contenant du texte dactylographié. Le logiciel analyse l'image, corrige les éventuels imperfections, repère les zones de texte, procède à une reconnaissance des caractères et enfin restitue à l'utilisateur le texte reconnu dans un format éditable.

L'état de l'art

La première machine d'OCR fut créée par Gustav Tauschek, un ingénieur allemand, en 1929. Elle contenait un détecteur photosensible qui pointait une lumière sur un mot lorsqu'il correspondait à un gabarit contenu dans sa mémoire. Par la suite, David Shepard avec l'aide de son ami, Harvey Cook, décidèrent sous la demande de Frank Rowlett de construire une machine permettant la conversion de messages imprimés en langage machine pour le traitement informatique. Cette machine fût baptisé "Gizmo", Shepard fonda alors Intelligent Machines Research Corporation (IMR), qui livra les premiers systèmes d'OCR au monde exploités par des sociétés privées. Elle fournit les premières machines d'OCR à Reader's Digest en 1955 puis à l'Ohio Bell Telephone Company et à l'US Air Force. Depuis les OCR sont assez répandu et sont beaucoup utilisé notamment par les systèmes postaux de nombreux pays afin de reconnaître automatiquement une adresse et de trier le courrier.

Jusqu'à une date récente, le fonctionnement des systèmes d'OCR performants était peu connu car protégé par le secret industriel ; les logiciels open-source disponibles (ex : GOcr) étant plutôt l'œuvre d'amateurs. La publication en open-source de systèmes performants (en particulier Tesseract en 2006) a quelque peu changé cette situation. La plupart des OCR opèrent de la façon suivante : pré-analyse de l'image, segmentation en lignes/caractères, reconnaissance, post-traitement (correction de mots, etc..) et enfin conversion en format texte. La reconnaissance peut se faire de trois manières :

1. Par classification par caractéristiques (Features) : faisant appel à un réseau de neurones.
2. Par méthodes métriques : on calcule la distance entre deux lettres.
3. Par méthodes statistiques : on utilise les chaînes de Markov.

Les logiciels les plus performants sont capable de détecter directement la police utilisée et ainsi d'éviter la phase d'apprentissage, si ce n'est pas le cas il faut d'abord fournir au logiciel les lettres utilisées par la police qui doit être reconnue. Parmi les logiciels les plus populaires on trouve aujourd'hui : GOCR (Unix,Windows), OCRopus (Unix) et Tesseract (Unix,Windows) qui sont tous les trois libres. Et d'autres logiciels propriétaires comme : Adobe Acrobat Professional (Windows, Mac OS), ExactScan (Mac OS), OCRkit (Mac OS) et Scansoft Omnipage (Windows).

Règles imposées

Le projet devra répondre au règles en vigueur sur le code à l'EPITA. D'autre part sera fournit à chaque soutenance : un plan de soutenance (comportant le déroulement prévu de la soutenance), un rapport de soutenance (d'une vingtaine de pages pour les deux premières soutenances, et de 50 pages à la soutenance finale) et un site web. A la soutenance finale le groupe fournira en plus des éléments précédents : un dossier d'exploitation (comportant un manuel d'installation et d'utilisation), le projet en lui même (avec ses sources et les bibliothèques utilisées) ainsi qu'une procédure d'installation/désinstallation.

3. Découpage du projet

Notre projet sera découpé de la manière suivante :

1. Le pré-traitement de l'image.
2. Le réseau de neurones.
3. L'apprentissage du réseau de neurone.
4. L'interface graphique.

3.1 Pré-traitement de l'image

Dans cette partie, il s'agit d'abord de préparer l'image afin qu'elle soit exploitable. En premier lieu, il convient d'appliquer lissage afin d'éviter tous les bruits qui pourraient venir contrarier la reconnaissance des caractères. Il faut cependant faire attention à ne pas enlever certains "bouts de caractères" comme l'accent sur le é, le point sur le i par exemple.

Ensuite, il convient de convertir l'image en noir et blanc, ou du moins d'augmenter le contraste afin de faire ressortir de la manière la plus nette possible la différence entre les caractères et le fond de l'image.

Par la suite, il est peut être nécessaire de recadrer l'image, voir de la redresser afin de mieux cibler la zone de texte dans laquelle il va falloir extraire les caractères.

Enfin, une fois la zone de texte correctement détectée est mise en évidence, il faut détecter les lignes, afin de les prendre une par une, puis découper ces lignes en mots, pour finalement découper tous les mots en caractères. Les caractères seront alors transmis un par un au réseau de neurone.

Cette partie sera codée en C.

3.2 Réseau de neurones

Parmi les trois méthodes de reconnaissance cités plus haut, nous avons fait le choix de la méthode par classification par caractéristiques (Features). Cette méthode fait donc appel à un réseau de neurone.

Les neurones dits formels sont en fait une représentation simplifiée des neurones biologiques. Ces réseaux de neurones formels simples peuvent théoriquement réaliser des fonctions logiques, arithmétiques et symboliques complexes.

Le neurone formel est conçu comme un automate doté d'une fonction de transfert qui transforme ses entrées en sortie selon des règles précises. Ces neurones sont par ailleurs associés en réseaux dont la topologie des connexions est variable : réseaux proactifs, récurrents, etc. Enfin, l'efficacité de la transmission des signaux d'un neurone à l'autre peut varier : on parle de « poids synaptique », et ces poids peuvent être modulés par des règles d'apprentissage.

Cette partie sera donc certainement codée en Objective Caml afin de pouvoir définir un neurone comme un objet.

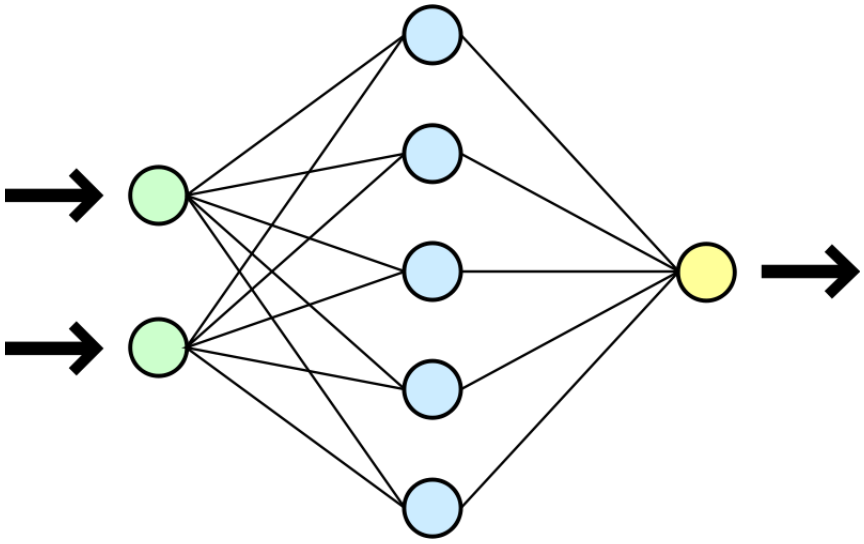

FIGURE 3.1 – Schéma simplifié d'un réseau de neurone

3.3 Apprentissage du réseau de neurone

Ce réseau de neurones formels va nous permettre d'opérer rapidement à des classifications et d'apprendre à les améliorer. Les paramètres importants de ce modèle sont les coefficients synaptiques et le seuil de chaque neurone, et la façon de les ajuster. Ce sont eux qui déterminent l'évolution du réseau en fonction de ses informations d'entrée. Il faut choisir un mécanisme permettant de les calculer et de les faire converger si possible vers une valeur assurant une classification aussi proche que possible de l'optimale. C'est ce qu'on appelle la phase d'apprentissage du réseau. Dans un modèle de réseaux de neurones formels, apprendre revient donc à déterminer les coefficients synaptiques les plus adaptés.

Il est alors question de fournir au réseau des modèles considérés comme parfaits des caractères. Ces modèles vont donc varier en fonction de la police à reconnaître. Cette police devra donc être spécifiée par l'utilisateur afin que le réseau de neurone puisse apprendre sur un modèle prédéfini, et qu'il puisse procéder à la reconnaissance dans de bonnes conditions. Le principe est simple, on présente une image parfaite d'un caractère au réseau, en lui précisant quel est ce caractère, celui-ci commence alors à les classifier.

Cette partie sera codée en C et/ou Ocaml suivant les besoins.

3.4 Interface Graphique

L'interface se veut claire et fonctionnelle, elle doit permettre tout d'abord à l'utilisateur de sélectionner un fichier image avec le texte à reconnaître. Elle sera divisée en deux parties : une partie avec l'image de base et les éventuels traitements qu'elle subit, et une seconde partie contenant le texte éditable reconnu dans une text-box.

L'interface permettra également de choisir les diverses options de traitement, et proposera à l'utilisateur d'enregistrer le résultat de la reconnaissance dans un fichier. Il sera bien sûr possible de choisir une nouvelle image à tout moment afin de recommencer le traitement. L'utilisateur n'aura au final pas besoin d'utiliser la console pour tirer pleinement parti du logiciel. Une aide pourra également être affichée via l'interface.

Cette partie sera codée en C.

3.5 Site Web et Rapport

Le site web contiendra une page d'accueil avec différents liens vers :
- Une présentation du projet (historique, problèmes rencontrées...).
- Une présentation des membres.
- Une page de téléchargement du projet et du rapport.
- Les liens utiles au projet (bibliothèques, documentation,...)
- Une page de contact.

Les rapports de soutenance et le rapport de projet seront tous écrit avec LateX.

3.6 Bonus

De nombreux bonus sont envisageables si jamais l'équipe termine ses objectifs en avance, notamment :
- Un post-traitement du texte (comparaison avec un dictionnaire de mots connus).
- Une reconnaissance de l'écriture manuscrite.
- Un apprentissage automatique en détectant la police automatiquement.
- La possibilité de fusionner plusieurs images de texte afin de fournir un seul document.
- La prise en charge de format texte enrichi pour l'enregistrement du texte.

Nous aurons sûrement d'autres idées au cours du projet, si nous avons les moyens de les réaliser et quelles nous semblent plus intéressantes que celles-ci nous les ajouterons au projet de nous même.

4. Calendrier des soutenances

Voici la répartition des tâches en fonction des membres du groupe et de la soutenance :

Légende :

\+ : Ebauche

++ : Avancé

+++ : Terminé

Première Soutenance	Aldéric	Julien	Romain	Benoit
Pré-traitement de l'image	+		+	
Réseau de neurones		+		+
Apprentissage		+		+
Interface graphique	+			
Site Web			++	
Rapport		+++		

Deuxième Soutenance	Aldéric	Julien	Romain	Benoit
Pré-traitement de l'image	++		++	
Réseau de neurones		++		++
Apprentissage		++		++
Interface graphique	+			
Site Web			+++	
Rapport		+++		

Soutenance Finale	Aldéric	Julien	Romain	Benoit
Pré-traitement de l'image	+++		+++	
Réseau de neurones		+++		+++
Apprentissage		+++		+++
Interface graphique	+++			
Site Web			+++	
Rapport		+++		

Cette répartition des tâches sera éventuellement réadaptée si un des membres du groupe finit sa partie plus vite que prévu. Il pourra ainsi aider les autres. Si tous les objectifs sont réalisés en avance, la répartition des tâches pourra être revu afin d'y intégrer des bonus. Il est important de noter que les rapports de soutenance et le rapport de projet, sont réalisés par tous les membres du groupe, la tâche est cependant assigné à Julien car il réalisera la structure du document Latex et fusionnera le texte de tous les membres du groupe.